

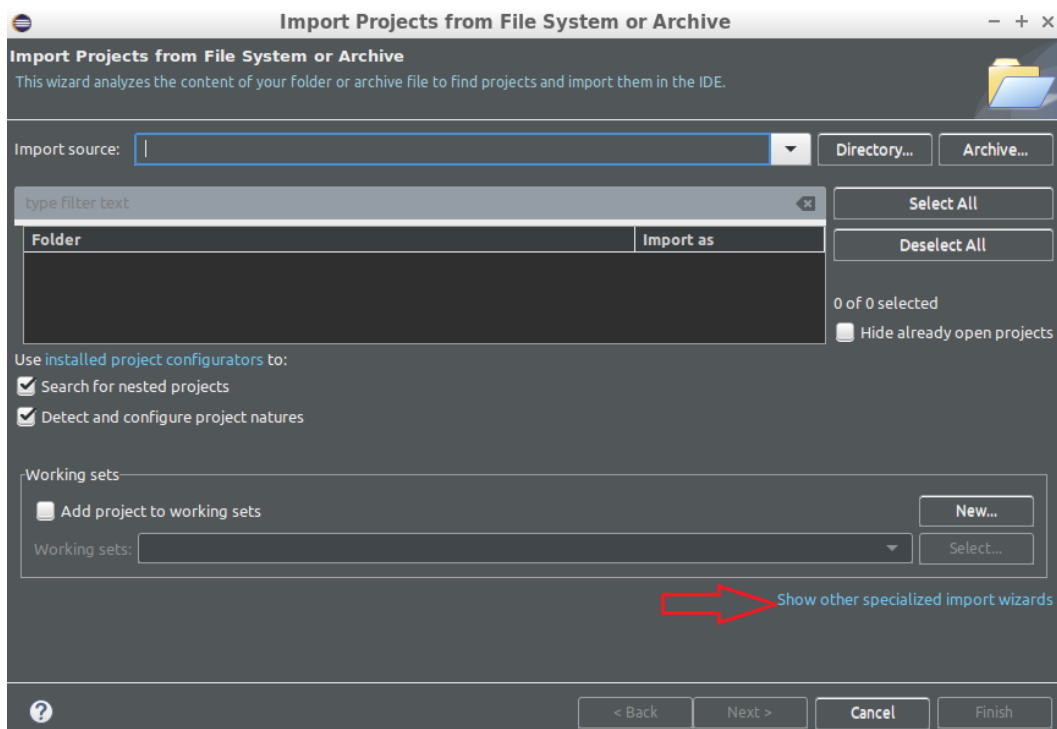
Configurar entorno para el Uso de la biblioteca compartida

Configuración en Eclipse:.....	1
1. Importar la Biblioteca:	1
2. En los Proyectos que utilizan la biblioteca:	3
3. Para correr los proyectos que usan la librería.....	5
Configuración del makefile del proyecto:.....	6
Configuración de la variable de Entorno de la VM:	7
1. De forma temporal:	7
2. De forma permanente:	7

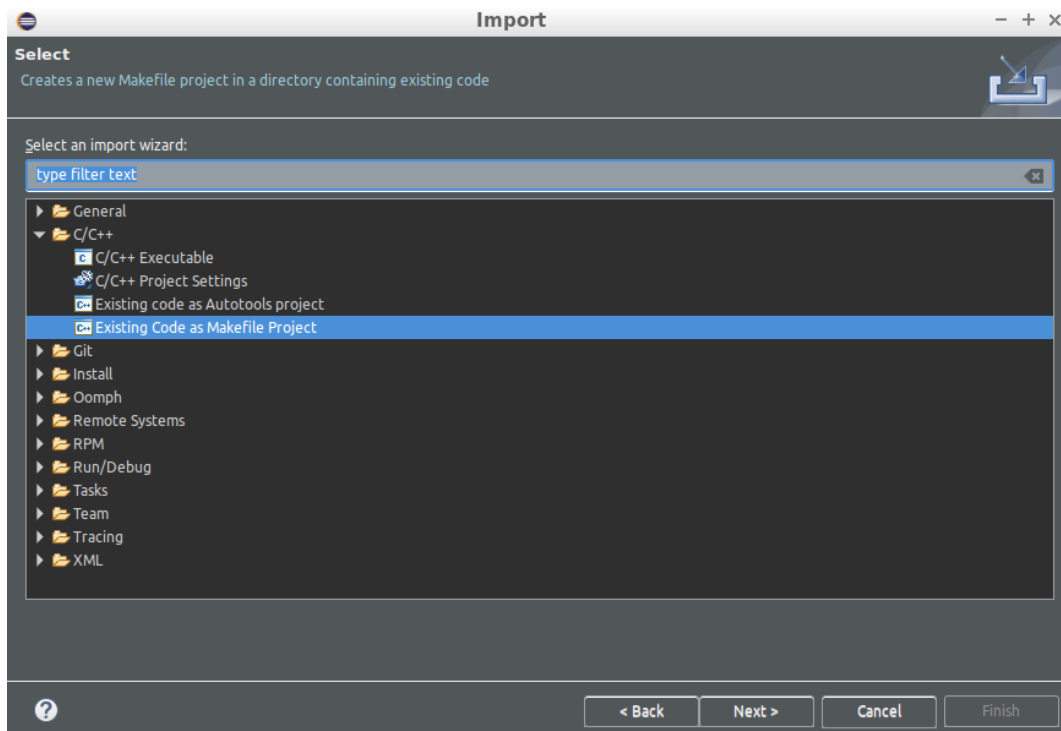
Configuración en Eclipse:

1. Importar la Biblioteca:

Acceder al Menu *File -> Open Projects from File System...* Y click *Show other specialized import wizards*



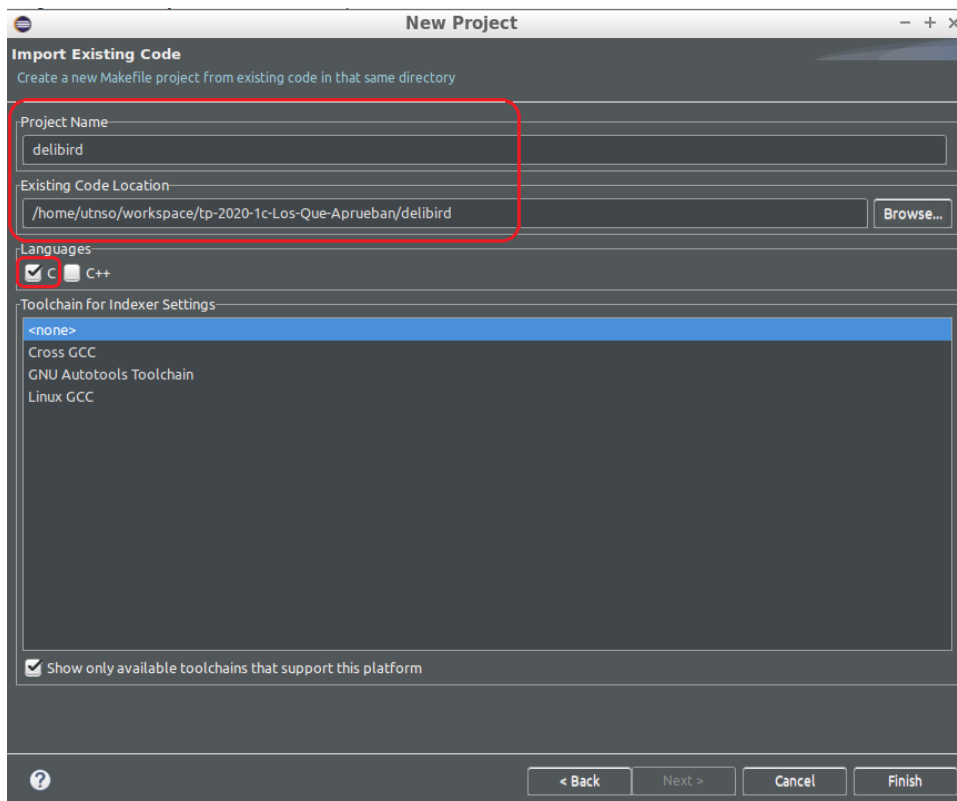
Seleccionamos *Existing Code as Makefile Project* y luego *Next*:



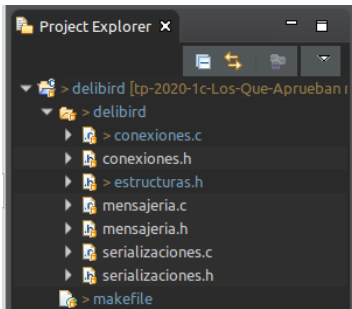
En *Languages* dejamos seleccionado solo **C**.

En *Existing code Location*, Click *Browse* y buscamos la carpeta raíz donde está la biblioteca y click *Ok*:

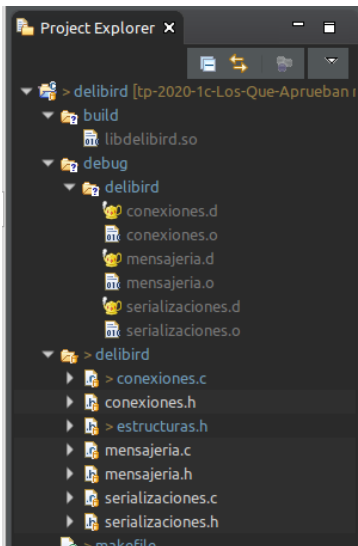
En *Project Name* se completará directamente:



Click *Finish*.



Probar compilación, seleccionamos el proyecto, Click derecho -> Build Project:

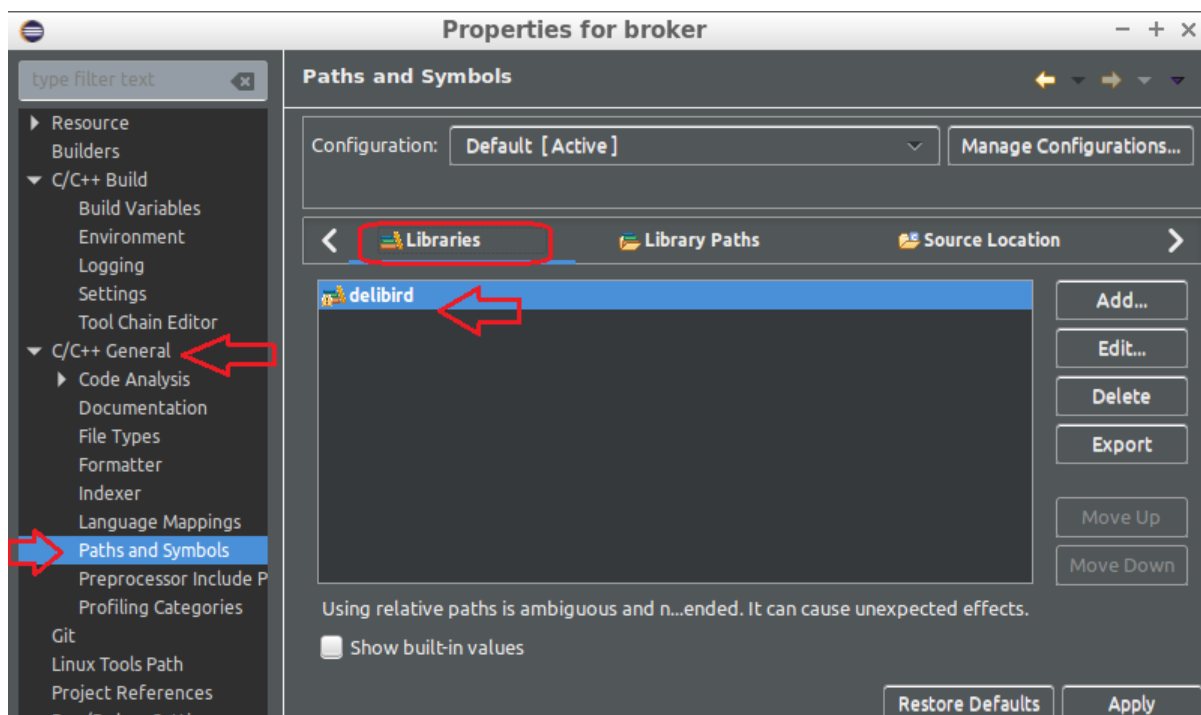


2. En los Proyectos que utilizan la biblioteca:

Seleccionar el Proyecto que use la biblioteca, Click derecho -> Propiedades (Alt + Enter):

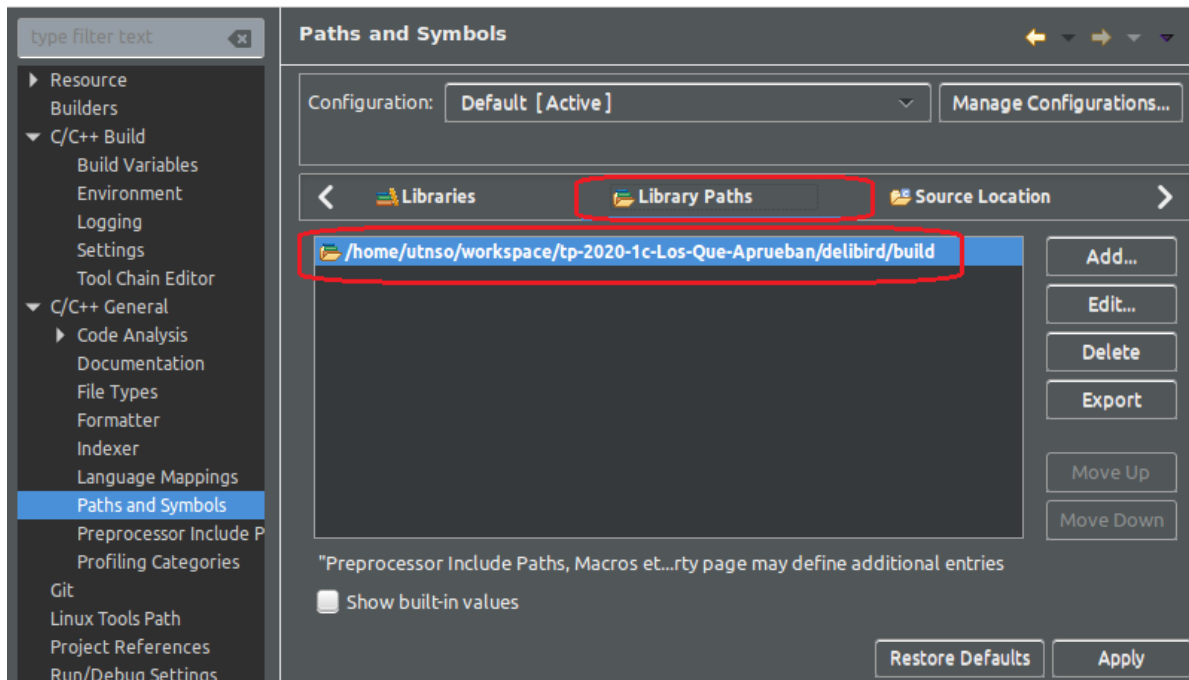
C/C++ General -> Path and Symbols:

Tab Libraries: Click *Add* -> delibird

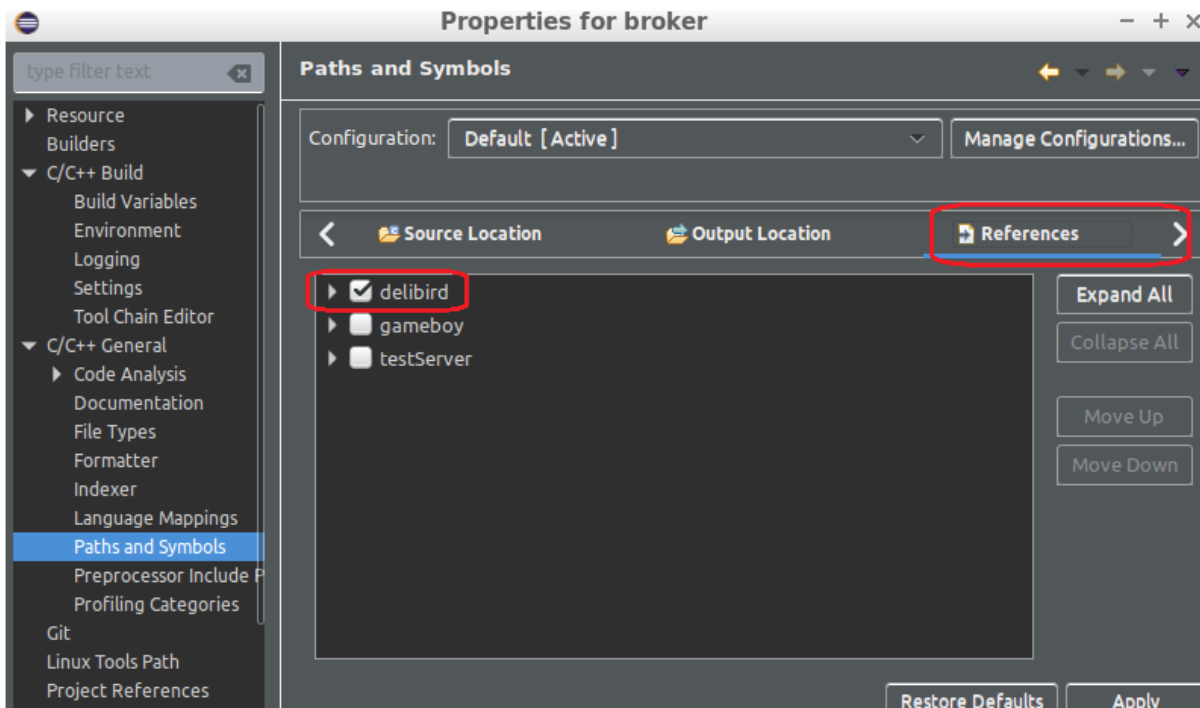


Tab **Library Paths** : Click Add. Completamos con la ruta completa donde está el archivo de compilación de la biblioteca libdelibird.so en el ejemplo es:

- **/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build**



Tab **References**: Seleccionar **delibird**

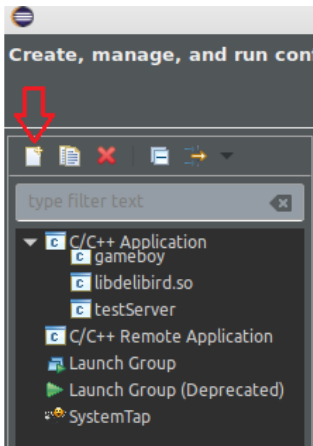


Click **Apply** and **Close**.

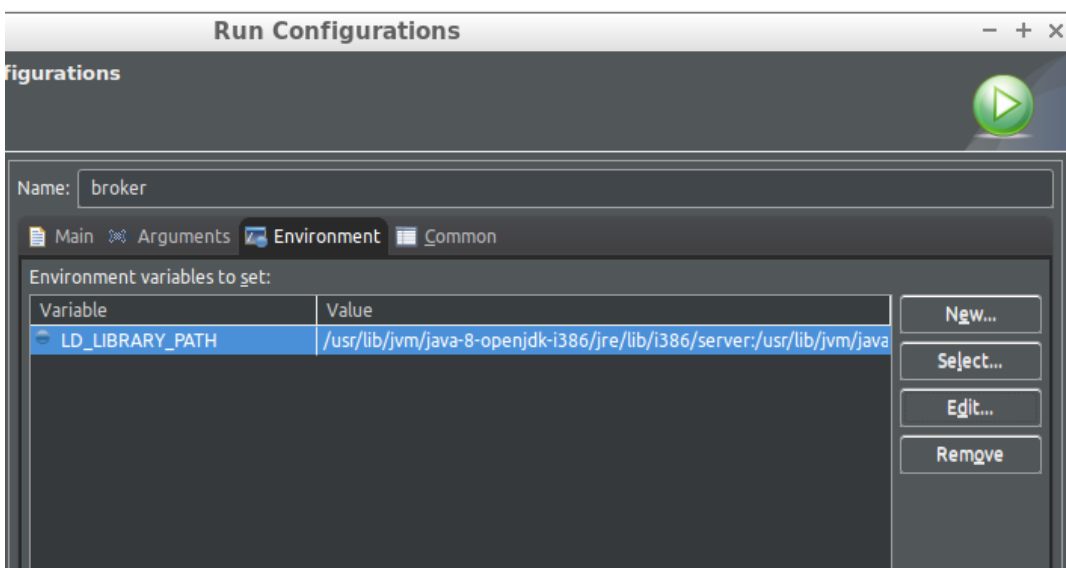
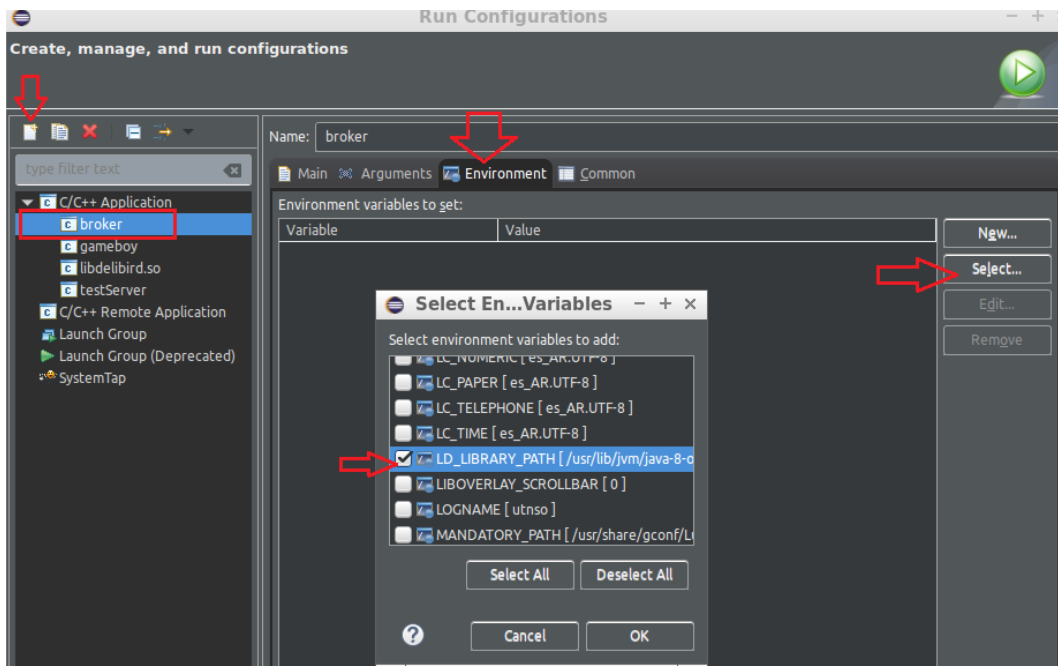
3. Para correr los proyectos que usan la librería.

Seleccionamos el proyecto, *Click Derecho -> Run as -> Run configurations...*

Si no está en la lista lo agregamos, Eclipse lo detecta solo:

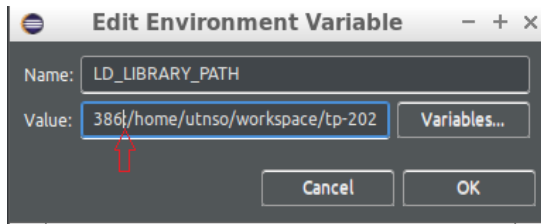


Tab Environment: Click *Select*. Seleccionamos **LD_LIBRARY_PATH** y Click *OK*:



Una vez que la agregamos, Click *Edit*: En *Value*: vamos al final del parámetro y agregamos de nuevo la ruta donde está el archivo de compilación de la biblioteca interponiendo ":"

- `:/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build`



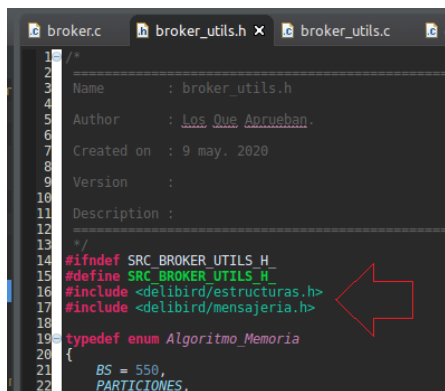
Click *Ok*. Luego *Apply*, por último *Close*.

Configuración del makefile del proyecto:

Para que los programas que utilizan esta biblioteca puedan usar las estructuras y funciones definidas, se necesita incluir los headers de la biblioteca incluyendo el nombre de la carpeta donde están: **delibird/**

```
#include <delibird/estructuras.h>
```

```
#include <delibird/headerXX.h>
```



En el archivo makefile:

- Por cada línea que el compilador GCC genera como salida un *archivo .o*, tenemos que permitir que gcc encuentre **todos los headers de la biblioteca**. Para ello anteponeamos **-I** ("i" latina Mayúscula) seguido de la ruta completa de la carpeta raíz donde esté la biblioteca, sin espacios. En el ejemplo:
 - **-I"/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird"**Al final de cada una de estas líneas agregamos **-l** ("L" minúscula) seguido del nombre de la librería, sin espacios:
 - **-ldelibird**
- En la línea que el compilador GCC genera como salida el archivo ejecutable (p.ej. broker), tenemos que permitir que gcc encuentre el archivo **.so** correspondiente a la biblioteca (en nuestro caso **libdelibird.so**). Para ello anteponeamos **-L** ("L" Mayúscula) seguido de la ruta completa de la carpeta raíz donde está el archivo de compilación, sin espacios. En el ejemplo:
 - **-L"/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build"**Al final de esta líneas agregamos **-l** ("L" minúscula) seguido del nombre de la librería, sin espacios:
 - **-ldelibird**

```
all: clean broker

broker:
gcc -I"/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird" -Wall -Werror -g -fPIC src/broker_utils.c -c -o debug/broker_utils.o -lcommons -ldelibird
gcc -I"/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird" -Wall -Werror -g -fPIC src/broker.c -c -o debug/broker.o -lcommons -ldelibird
gcc -L"/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build" -Wall -Werror -g debug/broker.o debug/broker_utils.o -o bin/broker -lcommons -ldelibird

clean:
rm -f ./debug/*.o ./bin/broker
```

Configuración de la variable de Entorno de la VM:

1. De forma temporal:

En la terminal donde vamos a ejecutar y compilar nuestros programas que utilicen la biblioteca corremos:

- **export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/directorio_donde_está_el_archivo_"biblioteca.so"**

Solo la ruta, no se incluye el archivo.so

```
utnso@lubuntuSO: ~  
utnso@lubuntuSO:~$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build  
utnso@lubuntuSO:~$
```

Probamos compilar y correr el programa:

```
utnso@lubuntuSO: ~/workspace/tp-2020-1c-Los-Que-Aprueban/broker  
utnso@lubuntuSO:~/workspace/tp-2020-1c-Los-Que-Aprueban/broker$ make all  
rm -f ./debug/*.o ./bin/broker  
gcc -I"/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird" -Wall -Werror -g -fPIC src/broker_utils.c -c -o debug/broker_utils.o -lcommons -ldelibird  
gcc -I"/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird" -Wall -Werror -g -fPIC src/broker.c -c -o debug/broker.o -lcommons -ldelibird  
gcc -L"/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build" -Wall -Werror -g debug/broker.o debug/broker_utils.o -o bin/broker -lcommons -ldelibird  
utnso@lubuntuSO:~/workspace/tp-2020-1c-Los-Que-Aprueban/broker$ bin/broker  
[INFO] 13:23:05:439 BROKER/(17324:17324): Algoritmo Memoria: BS = 550  
[INFO] 13:23:05:439 BROKER/(17324:17324): Cant Elem: 8  
[INFO] 13:23:05:439 BROKER/(17324:17324): [28|43|72|79|23|70|55|39]  
utnso@lubuntuSO:~/workspace/tp-2020-1c-Los-Que-Aprueban/broker$
```

2. De forma permanente:

Editamos el archivo **.profile** que está en la carpeta raíz del perfil del usuario **utnso**

- **nano /home/utnso/.profile**

```
utnso@lubuntuSO: ~/workspace/tp-2020-1c-Los-Que-Aprueban/broker  
utnso@lubuntuSO:~/workspace/tp-2020-1c-Los-Que-Aprueban/broker$ nano /home/utnso/.profile
```

```
GNU nano 2.5.3 Archivo: /home/utnso/.profile  
~/.profile: executed by the command interpreter for login shells.  
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login  
# exists.  
# see /usr/share/doc/bash/examples/startup-files for examples.  
# the files are located in the bash-doc package.  
  
# the default umask is set in /etc/profile; for setting the umask  
# for ssh logins, install and configure the libpam-umask package.  
#umask 022  
  
# if running bash  
if [ -n "$BASH_VERSION" ]; then  
    # include .bashrc if it exists  
    if [ -f "$HOME/.bashrc" ]; then  
        . "$HOME/.bashrc"  
    fi  
fi  
  
# set PATH so it includes user's private bin directories  
PATH="$HOME/bin:$HOME/.local/bin:$PATH"  
  
# Add RVM to PATH for scripting. Make sure this is the last PATH variable change.  
export PATH="$PATH:$HOME/.rvm/bin"  
[[ -s "$HOME/.rvm/scripts/rvm" ]] && source "$HOME/.rvm/scripts/rvm" # Load RVM into a shell session *as a function*
```

En la última línea agregamos el comando "export" como explicamos en el punto anterior:

- **export LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build**

```
GNU nano 2.5.3 Archivo: /home/utnso/.profile Modificado
# ~/.profile: executed by the command interpreter for login shells.
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
# exists.
# see /usr/share/doc/bash/examples/startup-files for examples.
# the files are located in the bash-doc package.

# the default umask is set in /etc/profile; for setting the umask
# for ssh logins, install and configure the libpam-umask package.
#umask 022

# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin directories
PATH="$HOME/bin:$HOME/.local/bin:$PATH"

# Add RVM to PATH for scripting. Make sure this is the last PATH variable change.
export PATH="$PATH:$HOME/.rvm/bin"

[[ -s "$HOME/.rvm/scripts/rvm" ]] && source "$HOME/.rvm/scripts/rvm" # Load RVM into a shell session *as a function*

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build
```

Para Guardar los cambios: **Ctrl + O** y **Enter** para confirmar el nombre del archivo

```
Nombre del archivo a escribir: /home/utnso/.profile
^G Ver ayuda      M-D Format DOS   M-A Añadir       M-B Respalda fich
^C Cancelar       M-M Format Mac   M-E Anteponer    ^T A Ficheros
```

Por último **Ctrl + X** para salir.

Para verificar los cambios:

Cerramos la terminal. Luego abrimos otra nueva terminal y tipeamos:

- **env | grep LD**

```
utnso@lubuntuSO: ~/workspace/tp-2020-1c-Los-Que-Aprueban/broker
utnso@lubuntuSO:~/workspace/tp-2020-1c-Los-Que-Aprueban/broker$ env | grep LD
OLDPWD=/home/utnso
LD_LIBRARY_PATH=/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build:/home/utnso/workspace/tp-2020-1c-Los-Que-Aprueban/delibird/build
utnso@lubuntuSO:~/workspace/tp-2020-1c-Los-Que-Aprueban/broker$
```

Se debe verificar que la variable se mantuvo. Probar la compilación y correr los programas como en el punto anterior.